SOCIAL DATA SCIENCE

Text as Data

Sebastian Barfort

August 18, 2016

University of Copenhagen
Department of Economics

TEXT ALL OVER THE PLACE

The widespread use of the Internet has led to an astronomical amount of digitized textual data accumulating every second through email, websites, and social media. The analysis of blog sites and social media posts can give new insights into human behaviors and opinions. At the same time, large-scale efforts to digitize previously published articles, books, and government documents have been underway, providing exciting opportunities for social scientists.

Imai (2016).

We need to learn how to think about and work with these kinds of new data

RESOURCES

Names (selective): Will Lowe, Justin Grimmer, Kenneth Benoit, Margaret E. Roberts, Sven-Oliver Proksch, Suresh Naidy

R packages: `tm`, `quanteda`, `stm`, `stringr`, `tidytext`

IS YOUR FAVORITE PROFESSOR BIASED?

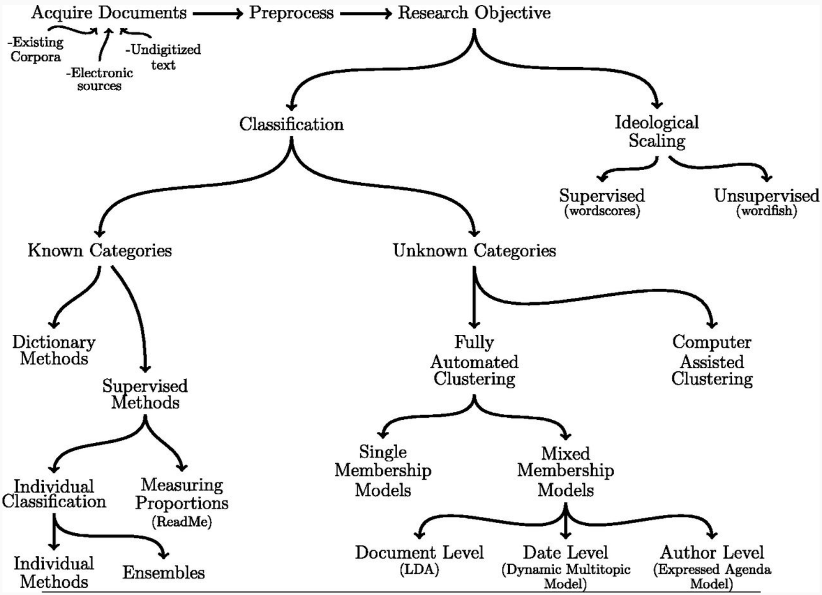
Jelveh, Zubin, Bruce Kogut, and Suresh Naidu. "Detecting Latent Ideology in Expert Text: Evidence From Academic Papers in Economics". EMNLP. 2014.

> *Previous work on extracting ideology from text has focused on domains where expression of political views is expected, but it's unclear if current technology can work in domains where displays of ideology are considered inappropriate. We present a supervised ensemble* n*-gram model for ideology extraction with topic adjustments and apply it to one such domain: research papers written by academic economists. We show economists' political leanings can be correctly predicted, that our predictions generalize to new domains, and that they correlate with public policy-relevant research findings. We also present evidence that unsupervised models can underperform in domains where ideological expression is discouraged.*

VALIDATION

> *We emphasize that the complexity of language implies that automated content analysis methods will never replace careful and close reading of texts. Rather, the methods that we profile here are best thought of as* amplifying *and* augmenting *careful reading and thoughtful analysis. Further, automated content methods are incorrect models of language. This means that the performance of any one method on a new data set cannot be guaranteed, and* ***therefore validation is essential when applying automated content methods****.*

Grimmer and Stewart (2013).


Acquire Documents
-Existing Corpora
-Electronic sources
-Undigitized text
Preprocess
Research Objective
Classification
Ideological Scaling
Supervised (wordscores)
Unsupervised (wordfish)
Known Categories
Unknown Categories
Dictionary Methods
Supervised Methods
Individual Classification
Measuring Proportions (ReadMe)
Individual Methods
Ensembles
Fully Automated Clustering
Computer Assisted Clustering
Single Membership Models
Mixed Membership Models
Document Level (LDA)
Date Level (Dynamic Multitopic Model)
Author Level (Expressed Agenda Model)

BAG OF WORDS

Bag of Words: The ordering and grammar of words does not inform the analysis.

Easy to construct sample sentences where word order fundamentally changes the nature of the sentence, but for most common tasks like measuring sentiment, topic modeling, etc. they do not seem to matter (Grimmer and Stewart 2013)

PRE-PROCESSING

Stemming: Dimensionality reduction. Removes the ends of words to reduce the total number of unique words in the data.

Ex: `family`, `families`, `families'`, etc. all become `famili`.

Stop words: Words that do not convey meaning but primarily serve grammatical purposes.

Uncommon Words: Typically, words that appear very often or very rarely are excluded.

Also typically discard punctuation (although not always!), capitalization, etc.

Classifying Documents into Known Categories

INTRODUCTION

Inferring and assigning text to categories is perhaps most common use of contant analysis in the social sciences

Ex: Classifying ads as positive/negative, is legislation about enviorenment, etc.

Two broad approaches:

Dictionary Methods: Use relative frequency of *key words* to measure presence of category in a given text

Supervised Learning: Build on and extend familiar manual coding tasks using algorithms

DICTIONARY METHODS

Perhaps the most simple and intuitive automated text classification method

Use the rate at which key words appear in a text to classify documents into categories or to measure extent to which documents belong to particular category

Dictionary: a list of words that classify a particular collection of words

Note: For dictionary methods to work well, the scores attached to each words must closely align with how the words are used in a particular context

Dictionaries are rarely validated

SUPERVISED LEARNING METHODS

Dictionary methods require that we are able to apriori identify words that separate classes

This can be wrong and/or inefficient

Supervised learning models are designed to automate the hand coding of documents

Supervised learning models: Human coders categorize a set of documents by hand. The algorithm then “learns” how to sort the documents into categories using these *training data* and apply its predictions to new unlabeled texts

APPROACH

1. Construct a *training set*
2. Apply the supervised learning method using cross-validation
3. Decide on “best” model and classify the remaining documents

Classification with Unknown Categories

INTRODUCTION

Supervised and dictionary methods assume a well-defined set of categories

Often, this set of categories is difficult to derive beforehand

Is must be *discovered* from the text itself

Unsupervised Learning: Try to learn underlying features of text without explicitly imposing categories of interest

1. Estimate set of categories
2. Assign documents (or part of documents) to those categories

Often: topic-models

Measuring Latent Features in Texts

INTRODUCTION

Can we locate actors (politicians, newspapers, researchers) in an ideological space using text data?

Assumption: Ideological dominance. Actors' ideological preferences determine what they discuss in texts.

Wordscores: Supervised learning approach. Special case of dictionary method.

Wordfish: Unsupervised learning approach. *Discover* words that distinguish locations on a policy scale.

WORDSCORE

1. Select *reference* texts that define the position in the policy space (e.g. a conservative and liberal politician)
2. Use training data to determine relative frequency of words. Creates a measure of how well various words separate the categories
3. Use these word scores to scale remaining texts.

Disadvantage: Conflates policy dominance with stylistic differences

Predicting Yelp Reviews

INTRODUCTION

David Robinson: Does sentiment analysis work? A tidy analysis of Yelp reviews

Sentiment analysis is often used by companies to quantify general social media opinion (for example, using tweets about several brands to compare customer satisfaction).

One of the simplest and most common sentiment analysis methods is to classify words as “positive” or “negative”, then to average the values of each word to categorize the entire document.

Can we use this approach to predict Yelp reviews?

DATA

Can be downloaded from here

```r
library("readr")
library("dplyr")

infile = "../nopub/yelp_academic_dataset_review.json"
review_lines = read_lines(infile,
                          n_max = 50000,
                          progress = FALSE)
```

FROM JSON TO DATA FRAME

```
library("stringr")
library("jsonlite")

reviews_combined = str_c("[",
                         str_c(review_lines,
                               collapse = ", "),
                         "]")

reviews = fromJSON(reviews_combined) %>%
  flatten() %>%
  tbl_df()
```

TIDY TEXT DATA

Right now, there is one row for each review.

Remember bag of words assumption: predictors are at the word, not sentence level

-> We need to tidy the data

```r
library("tidytext")
review_words = reviews %>%
  select(review_id, business_id, stars, text) %>%
  unnest_tokens(word, text)

review_words %>% dim
```

```
## [1] 5930037       4
```

REMOVE STOP WORDS

```
review_words = review_words %>%
  filter(!word %in% stop_words$word) %>%
  filter(str_detect(word, ”^[a-z’]+$”))
```

review_id	business_id	stars	word
Ya85v4eqdd6k9Od8HbQjyA	5UmKMjUEUNdYWqANhGckJw	4	hoagie
Ya85v4eqdd6k9Od8HbQjyA	5UmKMjUEUNdYWqANhGckJw	4	institut
Ya85v4eqdd6k9Od8HbQjyA	5UmKMjUEUNdYWqANhGckJw	4	walkin
Ya85v4eqdd6k9Od8HbQjyA	5UmKMjUEUNdYWqANhGckJw	4	throwb
Ya85v4eqdd6k9Od8HbQjyA	5UmKMjUEUNdYWqANhGckJw	4	ago

AFINN

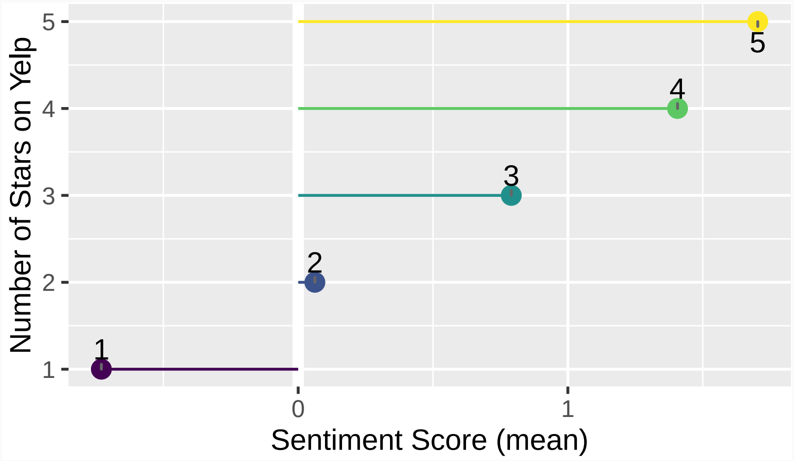
```
AFINN = sentiments %>%
  filter(lexicon == "AFINN") %>%
  select(word, afinn_score = score)
```

word	afinn_score
abandon	-2
abandoned	-2
abandons	-2
abducted	-2
abduction	-2

```
reviews_sentiment = review_words %>%
  inner_join(AFINN, by = "word") %>%
  group_by(review_id, stars) %>%
  summarize(sentiment = mean(afinn_score))
```

review_id	stars	sentiment
__-r0eC3hZlaejvuliC8zQ	5	4.0000000
__77nP3Nf1wsGz5HPs2hdw	5	1.6000000
__DK9Vsmyoo0zJQhIl5cbg	1	-2.1000000
__ELCJ0wzDM2QNRfVUq26Q	5	3.5000000
__esH_kgJZeS8k3i6HaG7Q	5	0.2142857

```r
df.review = reviews_sentiment %>%
  group_by(stars) %>%
  summarise(m.sentiment = mean(sentiment))
```

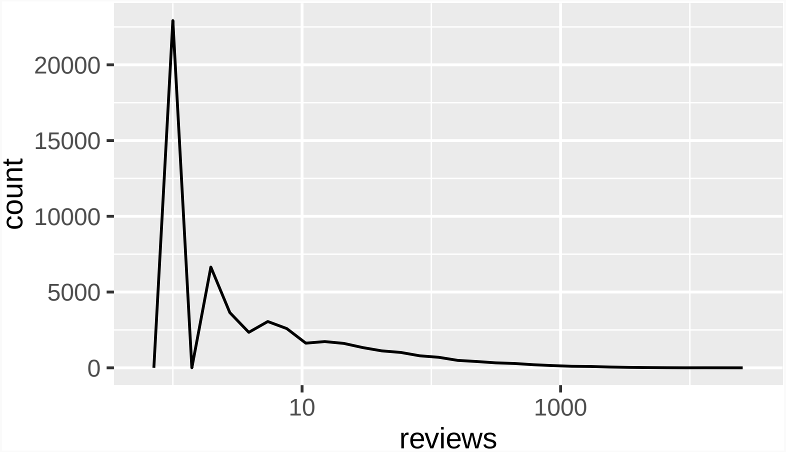

Number of Stars on Yelp
5
4
3
2
1
0
1
Sentiment Score (mean)
5
4
3
2
1

WORD FREQUENCY

```r
review_words_counted = review_words %>%
  count(review_id, business_id, stars, word) %>%
  ungroup()
```

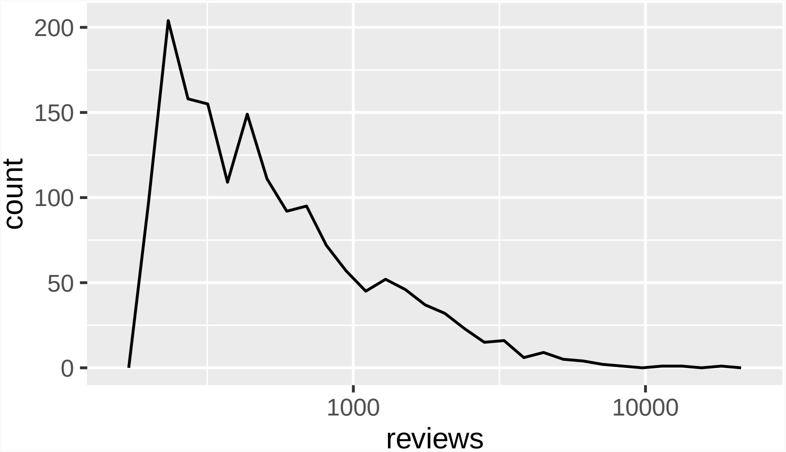
review_id	business_id	stars	word
__-r0eC3hZlaejvuliC8zQ	qemCNgjeYGcFsRwxW9x4xw	5	amazing
__-r0eC3hZlaejvuliC8zQ	qemCNgjeYGcFsRwxW9x4xw	5	attentive
__-r0eC3hZlaejvuliC8zQ	qemCNgjeYGcFsRwxW9x4xw	5	breakfast
__-r0eC3hZlaejvuliC8zQ	qemCNgjeYGcFsRwxW9x4xw	5	cheese
__-r0eC3hZlaejvuliC8zQ	qemCNgjeYGcFsRwxW9x4xw	5	chili

```
word_summaries = review_words_counted %>%
  group_by(word) %>%
  summarize(businesses = n_distinct(business_id),
            reviews = n(),
            uses = sum(n),
            average_stars = mean(stars)) %>%
  ungroup() %>%
  arrange(reviews)
```


count
20000
15000
10000
5000
0
10
1000
reviews

```r
word_summaries_filtered = word_summaries %>%
  filter(reviews >= 200, businesses >= 10)
```

word	businesses	reviews	uses	average_stars
lives	162	200	205	3.785000
regret	162	200	206	3.915000
bloody	80	201	246	3.621891
courses	84	201	242	3.800995
crowds	130	201	208	3.766169


200
150
100
50
0
count
1000
10000
reviews

POSITIVE WORDS

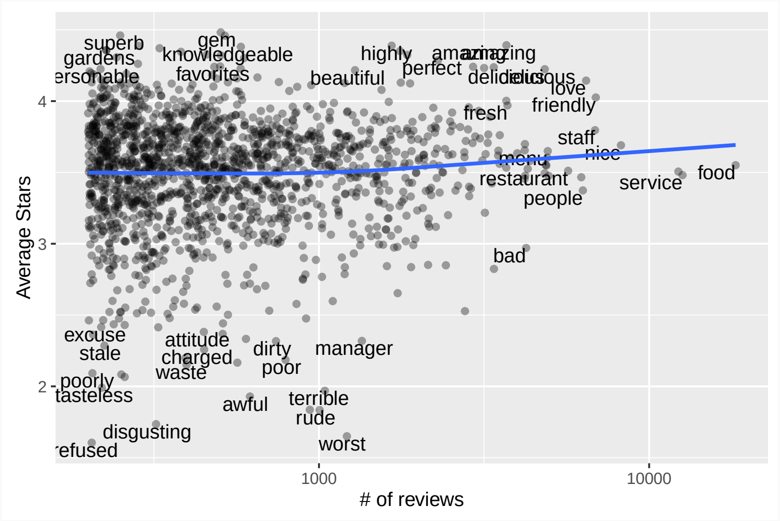
```
df.0 = word_summaries_filtered %>%
  arrange(-average_stars)
```

word	businesses	reviews	uses	average_stars
gem	272	504	509	4.482143
superb	171	250	253	4.460000
incredible	268	519	554	4.458574
amazing	927	3696	4240	4.391775
highly	736	1660	1729	4.388554

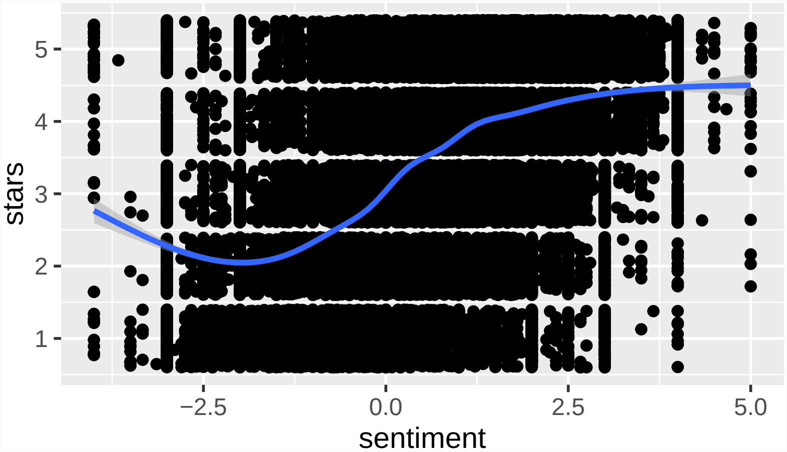
NEGATIVE WORDS

```
df.1 = word_summaries_filtered %>%
  arrange(average_stars)
```

word	businesses	reviews	uses	average_stars
refused	178	205	226	1.604878
worst	667	1215	1321	1.650206
disgusting	252	321	347	1.735202
rude	576	1005	1162	1.833831
horrible	582	938	1043	1.835821


superb
gardens
ersonable
gem
knowledgeable
favorites
highly
beautiful
amazing
amazing
perfect
delicious
delicious
love
friendly
fresh
staff
nice
menu
restaurant
people
service
food
bad
excuse
stale
attitude
charged
waste
dirty
poor
manager
poorly
tasteless
awful
terrible
rude
disgusting
refused
worst
4
3
2
Average Stars
1000
10000
# of reviews

STATISTICAL ANALYSIS



CROSS VALIDATION

```r
library("purrr")
library("modelr")
gen_crossv = function(pol,
                      data = reviews_sentiment){
  data %>%
    crossv_mc(200) %>%
    mutate(
      mod = map(train,
                ~ lm(stars ~ poly(
                  sentiment, pol),
                data = .)),
      rmse.test = map2_dbl(mod, test, rmse),
      rmse.train = map2_dbl(mod, train, rmse)
    )
}
```

```r
set.seed(3000)
df.cv = 1:8 %>%
  map_df(gen_crossv, .id = "degree")
```

Text Analysis of Donald Trump’s Tweets

```r
file = paste0("http://varianceexplained.org/",
              "files/",
              "trump_tweets_df.rda")
load(url(file))
```

QUESTION

David Robinson: Text analysis of Trump’s tweets confirms he writes only the (angrier) Android half

Who writes Donald Trump’s tweets?

They are written from two different devices: an iPhone and an Android

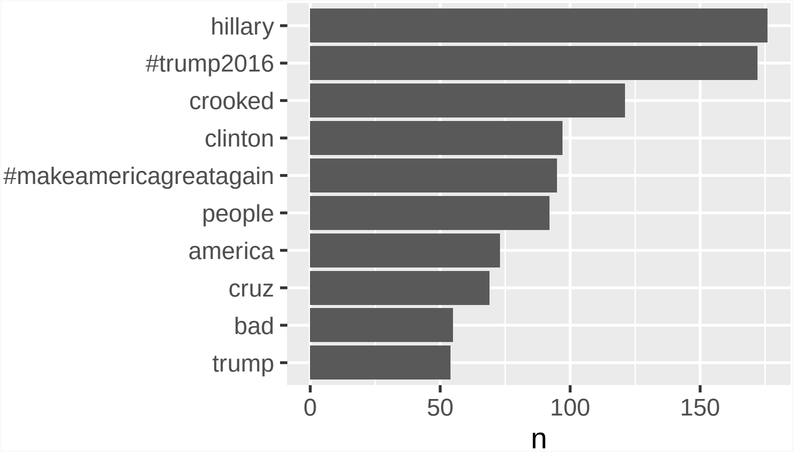
Can we examine quantitatively whether a tweet is written by Donal Trump himself or from someone on his staff?

```r
library("tidyr")

tweets = trump_tweets_df %>%
  select(id, statusSource, text, created) %>%
  extract(statusSource,
          "source", "Twitter for (.*?)<") %>%
  filter(source %in% c("iPhone", "Android"))
```

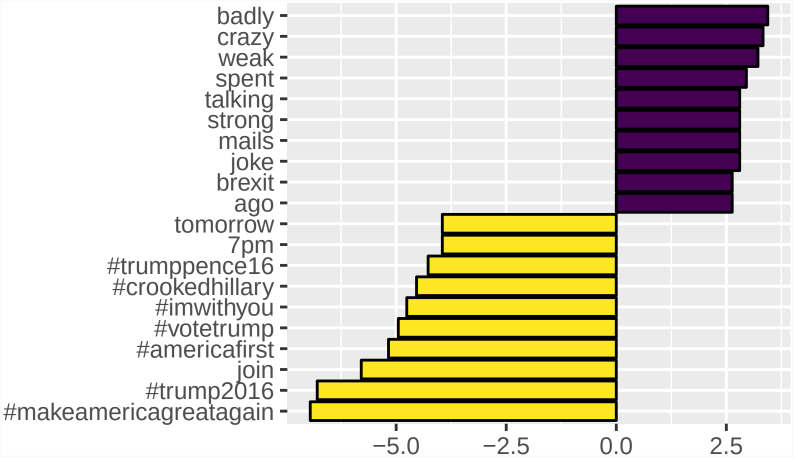
id	source	text
762669882571980801	Android	My economic policy speech will be carried
762641595439190016	iPhone	Join me in Fayetteville, North Carolina tom
762439658911338496	iPhone	#ICYMI: “Will Media Apologize to Trump?” h
762425371874557952	Android	Michael Morell, the lightweight former Act
762400869858115588	Android	The media is going crazy. They totally dist

```r
library("tidytext")
library("stringr")

reg = "([^A-Za-z\\d#@']|'(?![A-Za-z\\d#@]))"
tweet_words = tweets %>%
  filter(!str_detect(text, '^"')) %>%
  mutate(text =
           str_replace_all(text,
  "https://t.co/[A-Za-z\\d]+|&amp;", "")) %>%
  unnest_tokens(word, text, token = "regex",
                pattern = reg) %>%
  filter(!word %in% stop_words$word,
         str_detect(word, "[a-z]"))
```

id	source	created	word
676494179216805888	iPhone	2015-12-14 20:09:15	record
676494179216805888	iPhone	2015-12-14 20:09:15	health
676494179216805888	iPhone	2015-12-14 20:09:15	#makeamericagreatag
676494179216805888	iPhone	2015-12-14 20:09:15	#trump2016
676509769562251264	iPhone	2015-12-14 21:11:12	accolade


hillary
#trump2016
crooked
clinton
#makeamericagreatagain
people
america
cruz
bad
trump
0
50
100
150
n

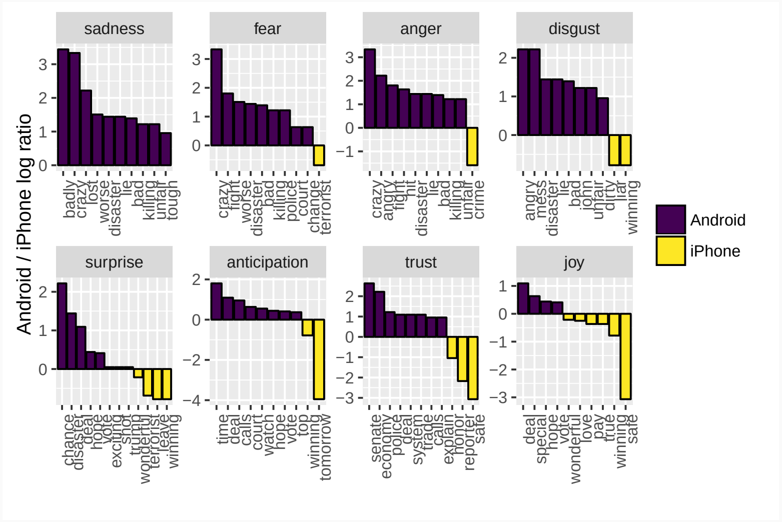
```r
android_iphone_ratios = tweet_words %>%
  count(word, source) %>%
  filter(sum(n) >= 5) %>%
  spread(source, n, fill = 0) %>%
  ungroup() %>%
  mutate_each(funs((. + 1) / sum(. + 1)), -word) %>%
  mutate(logratio = log2(Android / iPhone))
```


badly
crazy
weak
spent
talking
strong
mails
joke
brexit
ago
tomorrow
7pm
#trumppence16
#crookedhillary
#imwithyou
#votetrump
#americafirst
join
#trump2016
#makeamericagreatagain
−5.0
−2.5
0.0
2.5

SENTIMENT ANALYSIS

```
nrc = sentiments %>%
  filter(lexicon == ”nrc”) %>%
  select(word, sentiment)
```

word	sentiment
abacus	trust
abandon	fear
abandon	negative
abandon	sadness
abandoned	anger



sadness
fear
anger
disgust
surprise
anticipation
trust
joy
Android / iPhone log ratio
Android
iPhone
badly crazy lost worse disaster lie bad killing unfair tough
crazy fight worse disaster bad killing police court change terrorist
crazy angry fight hit disaster lie bad killing unfair crime
angry mess disaster lie bad john unfair dirty liar winning
chance disaster deal hope vote exciting shot trump wonderful terrorist leave winning
time deal calls court watch hope vote top winning tomorrow
senate economy police deal system trade calls explain honor reporter safe
deal special hope vote wonderful love pay true winning safe

YOUR TURN

Continue working with these data in groups

Think of interesting patterns you can explore in the data

What about the timing of tweets? Could everything be included in one large model?